

Faculdade de Engenharia da Universidade do Porto

Ano letivo 2015/2016



Pokemon Run

Laboratório de Computadores(LCOM)

MIEIC Turma 2 Grupo 12:

José Pedro Teixeira Monteiro up201406458@fe.up.pt

Pedro Duarte da Costa up201403291@fe.up.pt

Data de entrega: 4 de Janeiro de 2016

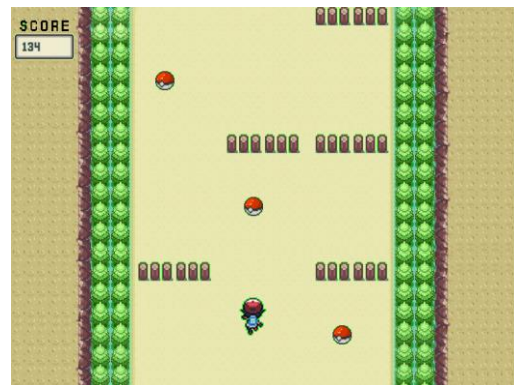
Índice

Instruções de utilizador -----	3
Dispositivos de entrada/saída -----	4
Organização de código -----	5
Gráfico de chamada das funções -----	8
Detalhes da implementação -----	9
Conclusões -----	10



Instruções de Utilizador

Ao iniciar o jogo é deparado com duas opções, "Play" e "Exit". Ao clicar em "Play" (com a tecla do lado esquerdo do rato), depara-se com a situação de jogo em que o objetivo é controlar o "Ash" (com as teclas A e D, A para andar para a esquerda e D para andar para a direita) de forma a que este não embata com os obstáculos que vão aparecendo e tentando apanhar o máximo de pokebolas possível. Quando o jogador embate num obstáculo termina a situação de jogo e é ilustrado um menu onde o jogador pode selecionar através do teclado, uma das três opções, ver o top score, jogar novamente ou sair do jogo.



Dispositivos de entrada/saída

Neste projeto foram desenvolvidos quase todos os laboratórios excepto o laboratório 7 (UART). Apresenta-mos, de seguida, uma tabela com a informação relativa a cada dispositivo utilizado.

Dispositivo	Utilização	Interrupções
Timer	Controlar o “frame rate”	SIM
Teclado	Situação de jogo	SIM
Rato	Selecionar uma opção no menu	SIM
Placa Gráfica	Menu, interface do jogo e score	NÃO
RTC	Associado a um score (quando foi realizado)	NÃO

Timer

Usamos as interrupções do timer para atualizar o jogo, a cada interrupção é realizado um update da situação em que se encontra, quer no menu, na situação de jogo ou visualização do score. Também é utilizado para o cálculo do score.

Teclado

Usado para leitura do keycode da tecla pressionada pelo jogador para a movimentação do “Ash”. Esta leitura é realizada no ficheiro player.c. Também é usado o teclado para quando o jogador perde, em que este é deparado com um novo menu onde pode ver o top score, jogar novamente ou sair do jogo.

Rato

Utilizado para a leitura dos packets do rato. Destes packets é utilizada a informação do deslocamento do rato e se é pressionada a tecla esquerda e direita do rato. Esta leitura é realizada no ficheiro MainMenuState.c.

Placa Gráfica

Utilizamos a placa gráfica (modo 0x114 com resolução 800x600) para a representação das várias situações do jogo. Foi utilizado o double buffering (sempre que é chamada a função refresh() representada no ficheiro interface.c), foram criadas algumas colisões assim como criadas algumas fontes.

RTC

Desenvolvemos o RTC de forma a obter a data e a hora para gravar o score do jogador.

Organização de Código

proj.c

Neste ficheiro é onde se dá a iniciação do jogo, é criado um objeto do tipo Game (explicado mais a frente), depois de criado é introduzido num ciclo onde este acaba quando é dada uma das condições de saída do jogo.

Membro responsável: Pedro Costa

Peso relativo: 6%

Game.h e Game.c

Módulo base do jogo, pois é aqui que são identificados as fases de jogo em que se encontra o jogador (menu, situação de jogo, visualização de score). Ao verificar a fase de jogo onde se encontra, é chamado o handler respetivo.

Membro responsável: Pedro Costa

Peso relativo: 10%

MainMenuState.h e MainMenuState.c

Módulo onde se encontra a construção do menu do jogo. Este menu possui a opção de jogar ou sair, isto sendo escolhido através do rato.

Membro responsável: Pedro Costa

Peso relativo: 6%

Button.h e Button.c

Neste módulo é onde se define as margens dos botões de jogo ("Play", "Exit", etc).

Membro responsável: Pedro Costa

Peso relativo: 2%

GameState.h e GameState.c

Neste módulo é onde se encontra toda a construção da situação de jogo. Onde é chamado as funções de movimento do background, dos obstáculos e das pokebolas, onde é atualizada a posição do "Ash" de acordo com as teclas pressionadas pelo jogador, etc.

Membro responsável: José Monteiro

Peso relativo: 15%

Bitmap.c e Bitmap.h

Módulo onde são definidas as funções que fazem o carregamento e desenharam na VRAM imagens em bmp.

Membro responsável: José Monteiro

Peso relativo: 4%

Collisions.h e Collisions.c

Módulo onde são definidas as funções que verificam as colisões realizadas em algumas situações do jogo.

Membro responsável: José Monteiro

Peso relativo: 6%

FinalState.h e FinalState.c

Neste módulo é onde esta organizado a situação depois de o jogador perder. Onde é apresentado o score, o top score do jogo e o jogador pode escolher sair do jogo ou jogar novamente.

Membro responsável: Pedro Costa

Peso relativo: 8%

fonts.h e fonts.c

Módulo onde são carregadas as “fonts” do jogo.

Membro responsável: Pedro Costa

Peso relativo: 3%

Obstacles.h e Obstacles.c

Neste módulo é onde são definidos os obstáculos, são umas barreiras que aparecem de forma random na situação do jogo e que se o jogador embater num destes obstáculos, perde. Estes obstáculos acompanham o movimento do background.

Membro responsável: José Monteiro

Peso relativo: 3%

Pokeball.h e Pokeball.c

Módulo muito semelhante ao modulo anterior referido (Obstacles.h e Obstacles.c). Muda apenas o facto de neste caso não se tratar de um obstáculo mas sim de uma pokebola em que quando existe uma colisão entre o “Ash” e uma pokebola, soma-se um valor constante ao score.

Membro responsável: José Monteiro

Peso relativo: 3%

Player.h e Player.c

Neste módulo encontra-se a construção do jogador, é onde se encontra as funções responsáveis pelo movimento e pela velocidade do “Ash”.

Membro responsável: José Monteiro

Peso relativo: 5%

RTC.h e RTC.c

Módulo onde foi implementado o rtc, que é utilizado para quando é realizado um score, associar a data de quando foi realizado.

Membro responsável: Pedro Costa

Peso relativo: 5%

Score.h e Score.c

Este módulo é responsável por gerar o score a cada jogada, sendo incrementado a cada interrupção e é somado ao score uma constante quando há uma colisão entre o “Ash” e uma pokebola.

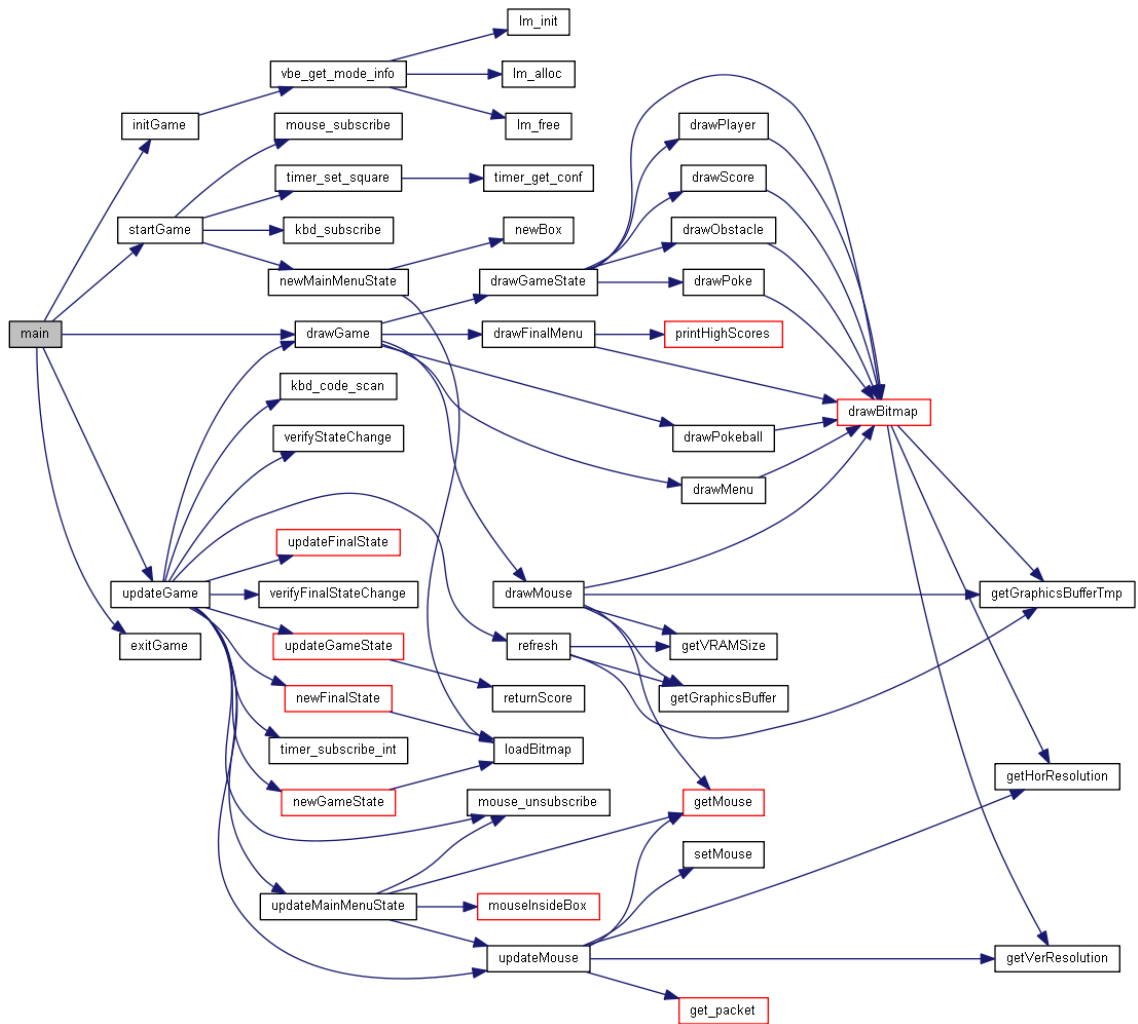
Membro responsável: José Monteiro

Peso relativo: 4%

Os módulos interface.h e interface.c, vbe.h e vbe.c, mouse.h e mouse.c, keyboard.h e keyboard.c e timer.h e timer.c são algumas partes do trabalho que foi realizado ao longo do semestre nas aulas laboratoriais, sendo responsável por ambos os membros.

Peso relativo: 20%

Gráfico de chamada das funções



Detalhes de implementação

Neste projeto fizemos uso das máquina de estado de forma a ter uma boa organização de código assim como um melhor redimento. Implementamos, por isso, três estados: 'MainMenuState', 'GameState' e 'FinalState'. Cada estado tem a sua condição de saída ou transição para outro estado. Para implementação da máquina de estados inspiramo-nos na explicação do estudante do MIEIC, Henrique Ferrolho, fornecida no seu blog: <http://difusal.blogspot.pt/>.

Na implementação da leitura e desenho dos bitmaps foi usado parte do código fornecido no seguinte site: <http://forums.fedoraforum.org/archive/index.php/t-171389.html>.

Fizemos uso da técnica de double buffering, em que todos os objetos são inicialmente desenhados num buffer de memória virtual e de seguida é copiado para a memória física da VRAM através da função disponível em linguagem C, memcpv().

O movimento do "Ash", é um movimento que possui aceleração, fornecendo um movimento mais "smooth" durante o jogo.

Ao longo da situação de jogo, vai sendo aumentada a velocidade de jogo (incrementando a deslocação do background, obstáculos e pokebolas). Este aumento é realizado passado x tempo, sendo o tempo obtido através das interrupções do timer (60 interrupções a cada segundo).

Os obstáculos e as pokebolas aparecem de forma "random", inicialmente são criadas três filas, com três objetos (esquerda, centro e direita). Estas filas tem uma distância vertical entre elas sempre constante e para cada fila, quando esta ultrapassa as dimensões do ecrã, é gerado novamente de forma random qual a posição dos três objetos da fila fica livre.

As colisões de jogo são verificadas através da posição dos objetos. Tanto para as pokebolas como para os obstáculos, cada um tem uma flag em que a colisão so é realizada quando as posições se intersejam e essa flag esta a 1, o que indica que quando foi gerado de forma random, este está a "visível".

Conclusões

Avaliação da unidade curricular

Ao longo do semestre houve uma pequena dificuldade ao início de cada lab pois achamos que a especificação não era a melhor, não sendo por vezes, muito objetiva. Contudo esta unidade curricular ensinou-nos a ser mais autónomos na aprendizagem de assuntos com os quais não estávamos familiarizados, como um sistema operativo ubuntu, aprendizagem mais profunda da linguagem C e sistemas de partilha de código (svn).

Contribuição no relatório: o relatório foi realizado em conjunto, tendo sido a realização deste uniforme por parte de ambos os elementos.

Instruções de utilização: apenas é necessário compilar (make) e de seguida, service run `pwd`/proj.